SOAR: VISUAL TOKEN PRUNING VIA CROSS-MODAL SUBSPACE OVERLAP GUIDANCE IN VLMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Abundant visual information strengthens vision-language model (VLM) perception, yet massive visual tokens raise inference costs. Existing visual token pruning methods have achieved significant progress by leveraging inter-token similarity or attention scores. However, such pruning guidance is largely restricted to local token-to-token similarity, ignoring the global geometric and structural relationships between the visual and textual subspaces. In this paper, we revisit VLM inference from a geometric perspective and present a novel guidance scheme for token compression. In particular, we identify a key observation: as LLM layers deepen, the subspaces spanned by the visual tokens and the textual tokens gradually align, and their overlap expands. To quantify this phenomenon, we propose Cross-Modal Subspace Overlap (CSO) to measure the extent to which visual tokens are spanned by text representations, revealing that more visual tokens in deeper layers can be linearly predicted by the text subspace. We accordingly propose Cross-Modal Residuals (CMR), which projects visual tokens onto the text subspace via Tikhonov-regularized least squares and exploits reconstruction residuals as an efficient geometric proxy to score tokens based on their unaligned features. Finally, based on CMR, we present SOAR (**S**ubspace **O**verlap and **A**lignment **R**esiduals), a training-free visual token pruning framework. Experiments on diverse VLM architectures verify the effectiveness of SOAR. For instance, on LLaVA-NeXT-7B, SOAR keeps only 11.1% of visual tokens while preserving 97.5% of the original average performance, achieving a $3.62\times$ prefill speedup, $2.49\times$ end-to-end speedup, and a $6.02\times$ KV-cache reduction.

1 INTRODUCTION

Vision Language Models (VLMs) have achieved remarkable progress in multimodal tasks such as visual question answering, image captioning, and document understanding. Representative models, including LLaVA (Liu et al., 2023), InternVL (Chen et al., 2024b), and Qwen-VL (Bai et al., 2023), typically encode images into numerous visual tokens and feed them into large language models together with text tokens for joint reasoning. However, the increasing number of visual tokens substantially raises self-attention computation, leading to higher inference latency and computational cost. This overhead is further amplified in high-resolution images, multi-image inputs, and video understanding. Therefore, compressing visual tokens while preserving model capability is crucial for efficient VLM inference.

Existing visual token compression methods mainly select tokens based on current layer attention (Chen et al., 2024a; Zhang et al., 2025b) or feature similarity (Bolya et al., 2023; Wen et al., 2025): the former characterizes the relevance between visual tokens and the current textual context, whereas the latter measures the redundancy among visual tokens. Although these signals provide useful criteria, they evaluate token importance from a strictly localized standpoint, leaving the global geometric alignment between modal subspaces unexplored. To address this limitation, we shift our focus from individual token correlations to the macroscopic representation geometry of VLMs. Specifically, we discover a critical property: as the network deepens, the feature space spanned by visual tokens undergoes a progressive structural convergence with the textual subspace, leading to a significantly enlarged overlapping region.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

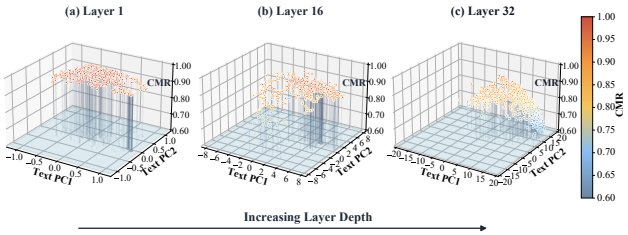


Figure 1: 3D visualization of visual tokens relative to the text subspace. Text tokens are centered and projected by PCA to obtain the first two principal directions, denoted as Text PC1 and Text PC2. Visual tokens are centered using the same text mean and projected onto these directions to obtain planar coordinates, while height and color indicate their CMR. More results are provided in Appendix A.

To quantify this phenomenon, we propose Cross-Modal Subspace Overlap (CSO) from a geometric representation perspective, which measures at the layer level the overall extent to which visual token representations are spanned by the text subspace, i.e., linearly predictable by the text context. While CSO characterizes this structural alignment trend at the layer level, token compression must operate on individual tokens. We therefore refine this perspective to the token level and propose Cross-Modal Residuals (CMR). Specifically, each visual token is projected onto the text subspace via Tikhonov-regularized least squares, and the reconstruction residual is utilized as an efficient geometric proxy to score tokens based on their unaligned features. A larger residual indicates that the token exhibits higher geometric independence from the text context, and thus has a higher retention necessity. This projection can be efficiently solved in closed form, introducing only minor additional overhead. As shown in Figure 1, visual token representations gradually move closer to the text subspace as layer depth increases, indicating that visual and textual representations do not evolve in isolated spaces, but progressively undergo structural convergence.

Building on the above analysis, we propose **SOAR** (**S**ubspace **O**verlap and **A**lignment **R**esiduals), a training-free visual token compression framework. SOAR selects tokens by harmonizing three complementary criteria: CMR for geometric non-redundancy, text-to-visual attention for task relevance, and residual-space diversity selection for representation coverage. Unlike de-duplication performed in the raw feature space, operating within the residual space attenuates text-aligned components through regularized reconstruction. Consequently, its directional variance more directly reflects the distinctive geometric characteristics of each token, enabling precise identification of genuinely complementary representations. SOAR is training-free and plug-and-play, making it directly applicable to diverse off-the-shelf VLMs.

In summary, our contributions are threefold:

- We reveal a novel representation phenomenon in VLMs: as network depth increases, visual and textual token subspaces undergo progressive structural convergence. Based on this, we introduce Cross-Modal Residuals (CMR) as an efficient geometric proxy for token pruning.
- We present SOAR, a training-free visual token pruning framework that synergistically integrates CMR, attention relevance, and residual-space diversity selection to balance geometric distinctiveness, task relevance, and representation complementarity.
- Extensive experiments across diverse VLM architectures and eight benchmarks demonstrate that SOAR substantially reduces computational and memory overhead while preserving baseline performance, delivering effective real-world inference acceleration.

2 Related Work

2.1 Multimodal Large Language Models

VLMs perform multimodal reasoning by feeding visual and text tokens into LLMs. Early systems use cross-attention or query-based designs, e.g., Flamingo (Alayrac et al., 2022), BLIP-2 (Li et al., 2023a), Kosmos-1 (Huang et al., 2023), and PaLM-E (Driess et al., 2023), followed by instruction-tuned models such as MiniGPT-4 (Zhu et al., 2024), mPLUG-Owl (Ye et al., 2023), LLaVA (Liu et al., 2023), and InstructBLIP (Dai et al., 2023). Recent high-resolution, general-purpose VLMs such as Qwen-VL (Bai et al., 2023), InternVL (Chen et al., 2024b), Gemini (Team et al., 2023), LLaVA-NeXT (Liu et al., 2024a), and Qwen2.5-VL (Bai et al., 2025) further improve OCR,

grounding, and visual reasoning, and extend to video understanding (Maaz et al., 2024; Lin et al., 2024; Li et al., 2024; 2025a). However, high-resolution images, multi-image inputs, and long videos rapidly inflate visual-token counts, incurring heavy self-attention and KV-cache overhead; low-loss visual token compression is thus crucial for their efficient inference.

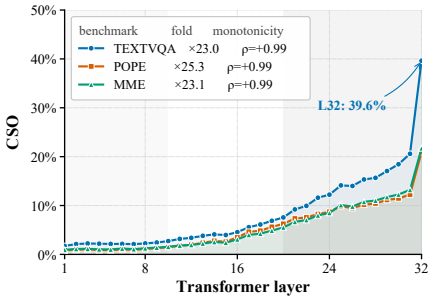


Figure 2: Layer-wise evolution of CSO in LLaVA-1.5-7B. CSO increases across Transformer layers.

2.2 Visual Token Compression for VLMs

Existing methods mainly fall into two categories. *Attention-score pruning* (Chen et al., 2024a; Zhang et al., 2025b; Xing et al., 2024; Huang et al., 2025) removes low-importance tokens without training, but attention scores are biased and unreliable indicators of true importance. *Similarity-based reduction* (Bolya et al., 2023; Wen et al., 2025; Yang et al., 2025b; Shang et al., 2025; Li et al., 2025b) merges similar tokens to preserve visual coverage, yet similarity need not correspond to text-query relevance. Other methods use diversity- and coverage-based selection (Alvar et al., 2025; Zhang et al., 2025a; Deng et al., 2025; Zou et al., 2025), adaptive token dropping (Chen et al., 2026), cross-modal information flow (Lin et al., 2025; Yang et al., 2025a) or objective-driven (Zhang et al., 2026). However, this line of guidance operates solely on local token-to-token similarity, overlooking the richer, global geometric relationships between visual and textual subspaces.

3 Analysis

Differing from these works, we revisit the VLM inference process and investigate the evolution of visual tokens and text tokens throughout this procedure. Figure 2 presents our observation.

We analyze all 32 layers of LLaVA-1.5-7B on three benchmarks, TextVQA (Singh et al., 2019), POPE (Li et al., 2023b), and MME (Fu et al., 2025), covering 16,284 samples. The results show that an increasing portion of visual-token representations can be explained by the text subspace, indicating that visual and textual representations undergo progressive structural convergence during forward propagation.

3.1 Measuring Cross Modal Subspace Overlap

To quantify this phenomenon, we define **Cross Modal Subspace Overlap (CSO)** to measure how much of the visual token representation at a given layer can be explained by the text subspace. The CSO at layer l is defined as

$$\mathrm{CSO} = 1 - \frac{\left\| V_c - \hat{V}_c \right\|_F^2}{\|V_c\|_F^2}. \tag{1}$$

Here, V_c denotes the matrix of all centered visual tokens at layer l (rather than an individual token); $\hat{V}_c$ its projection onto the text subspace; and $||\cdot||_F$ the Frobenius norm. CSO is a layer-level metric: a larger value means more visual information can be explained by the text subspace.

3.2 Visual Tokens Are Progressively Spanned by the Text Subspace

As shown in Figure 2, CSO consistently increases from shallow to deep layers across all three benchmarks: from 0.008 to 0.209 on POPE, from 0.009 to 0.217 on MME, and from 0.017 to 0.396 on TextVQA. Its Spearman correlation with layer depth reaches $\rho = +0.99$ on all three, indicating a highly consistent monotonic trend.

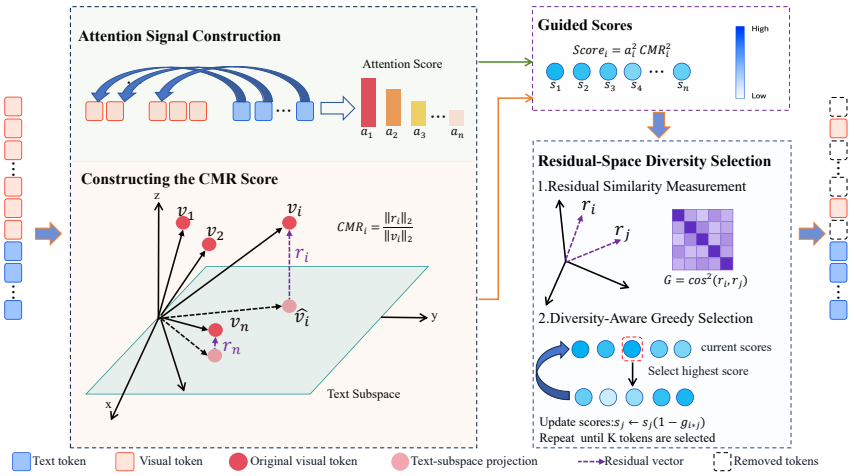


Figure 3: Overview of SOAR. SOAR measures each visual token's residual information with respect to the text subspace via CMR, combines it with attention relevance, and performs residual-space diversity selection to retain informative and non-redundant visual tokens for the compressed input sequence.

Note that the absolute value of CSO should not be read directly on the $[0, 1]$ scale. Since the number of text tokens is usually around $N_t \approx 30$, far smaller than the hidden dimension $D = 4096$, a random subspace can theoretically explain only about $N_t/D \approx 0.007$ of a high-dimensional representation. Thus the final-layer CSO of 0.209 on POPE is already about $29\times$ the random baseline, indicating that the explanatory power of the text subspace over visual representations is not random but gradually emerges as a cross-modal representation pattern in deeper layers. Details are provided in Appendix B.

Thus, visual representations do not evolve in geometric isolation across layers; instead, their feature space becomes increasingly spanned by the text subspace as network depth grows. This structural shift reflects the cross-modal representation convergence in VLMs, where textual feature spaces progressively align with visual directions through transformer layers. Crucially, this phenomenon unveils cross-modal structural redundancy in deeper layers: **a visual token whose representation is largely spanned by the text subspace exhibits high alignment redundancy, rendering it a prime candidate for token pruning.** Consequently, we can evaluate, at the token level, the geometric relationship between individual visual tokens and the text subspace—retaining tokens with high geometric independence while pruning those with negligible unaligned residuals. Accordingly, the next section formalizes the token-level Cross-Modal Residual (CMR) and presents our complete visual token compression framework, SOAR.

4 METHOD

Since CSO operates at the layer level while compression acts on individual tokens, we refine it to the token level and propose SOAR, as shown in Figure 3

4.1 CONSTRUCTING THE CMR SCORE

SOAR first computes a Cross-Modal Residual (CMR) score to quantify the feature component of each visual token that lies outside the text-spanned subspace: a lower score reflects stronger cross-modal alignment, whereas a higher score indicates greater geometric independence. At the l-th pruning layer, let $T \in \mathbb{R}^{N_t \times D}$ and $V \in \mathbb{R}^{N_v \times D}$ denote the text- and visual-token representation matrices, where D is the hidden dimension. SOAR first centers both modalities using the mean text representation:

$$\bar{t} = \frac{1}{N_t} \sum_{j=1}^{N_t} T_j, \quad T_c = T - \bar{t}, \quad V_c = V - \bar{t}. \tag{2}$$

Here $\bar{t}$ is the mean text representation, and T_c, V_c are the centered text and visual matrices. Using the text center as reference reduces the background components shared between the two modalities and highlights each visual token's unique deviation from the text subspace.

For the i-th centered visual token $v_{c,i}$, SOAR reconstructs it as a linear combination of text tokens, using a Tikhonov-regularized least-squares formulation for stability:

$$b_i = \arg\min_{b_i \in \mathbb{R}^{1\times N_t}} \|v_{c,i} - b_i T_c\|_2^2 + \lambda \|b_i\|_2^2. \tag{3}$$

Here b_i are the combination coefficients of $v_{c,i}$ over text tokens and λ is the regularization coefficient. The first term measures the reconstruction error from the text subspace, while the second constrains the coefficient magnitude to prevent unstable solutions caused by low-energy directions.

To adaptively determine λ, SOAR constructs the text Gram matrix:

$$G = T_c T_c^\top. \tag{4}$$

Here $G \in \mathbb{R}^{N_t \times N_t}$ characterizes the energy distribution of the text subspace, providing the basis for setting λ. Sorting its eigenvalues in descending order as $\sigma_1^2 \geq \cdots \geq \sigma_{N_t}^2$ and given an energy ratio threshold η, SOAR selects the smallest k satisfying

$$\sum_{j=1}^{k} \sigma_j^2 \geq \eta \sum_{j=1}^{N_t} \sigma_j^2. \tag{5}$$

Here η is the energy proportion to be covered and k is the number of dominant directions needed to reach it, delimiting the effective dimensionality of the text subspace. The regularization coefficient is then set as

$$\lambda = \sigma_k^2 + \epsilon, \tag{6}$$

where ϵ is a small constant for numerical stability. Setting λ to the k-th eigenvalue applies spectrum-dependent shrinkage to the text-subspace reconstruction. This yields the closed-form solution and the reconstruction:

$$b_i = v_{c,i} T_c^\top (G + \lambda I)^{-1}, \qquad \hat{v}_{c,i} = b_i T_c. \tag{7}$$

Here I is the identity matrix and $\hat{v}_{c,i}$ is the visual token reconstructed from the text subspace; the closed-form solution avoids iterative optimization and adds only negligible overhead. Finally, SOAR defines the CMR score as the ratio between the reconstruction residual and the original centered representation:

$$\mathrm{CMR}_i = \frac{\|v_{c,i} - \hat{v}_{c,i}\|_2}{\|v_{c,i}\|_2}. \tag{8}$$

Normalizing the residual by the token's original norm removes scale effects from varying feature magnitudes. A lower CMR_i denotes strong alignment with the text subspace and thus lower retention necessity, whereas a higher CMR_i preserves a larger unaligned geometric component, indicating higher geometric independence and preferential retention.

4.2 Attention Signal Construction

CMR measures the unique information in a visual token that cannot be explained by the text subspace, but geometric uniqueness does not necessarily indicate relevance to the current text context. Therefore, SOAR further introduces an attention signal to determine whether each visual token is actually attended to by the text tokens. For the i-th visual token, its text relevance score is defined as

$$a_i = \sum_{h \in \mathcal{H}_{\mathrm{top}}} \sum_{t \in \mathcal{T}} \alpha_{t,i}^h. \tag{9}$$

Here, $\mathcal{T}$ denotes the set of text tokens, and $\alpha_{t,i}^h$ denotes the attention weight from text token t to visual token i in the h-th attention head. To reduce the influence of noisy heads, SOAR keeps only the top ρ_h proportion of heads with the largest attention mass on the visual region, denoted as $\mathcal{H}_{\mathrm{top}}$. A larger a_i indicates stronger relevance to the current textual context. SOAR does not need to explicitly store the full attention matrix. Following the dual-flash attention strategy in SparseVLM (Zhang et al., 2025b), SOAR can compute the required attention signal while remaining compatible with FlashAttention, with details provided in Appendix C.

4.3 GUIDED SCORES

SOAR initializes each token's integrated score as

$$\mathrm{Score}_i = a_i^2 \, \mathrm{CMR}_i^2, \tag{10}$$

where a_i is the text-relevance (attention) score and CMR_i is the residual ratio unexplained by the text subspace; the product keeps attention from overlooking residual information and CMR from ignoring textual relevance.

4.4 RESIDUAL-SPACE DIVERSITY SELECTION

Given the guided scores defined above, SOAR aims to retain visual tokens that are both individually informative and mutually complementary. To this end, it performs residual-space diversity selection in two stages: it first measures the redundancy between visual tokens in the residual space, and then greedily selects tokens while progressively suppressing redundant candidates.

1. **Residual Similarity Measurement.** Selecting the top-K tokens by integrated score alone may retain tokens with similar residual directions, causing visual redundancy. To avoid this, SOAR measures token redundancy in the *residual space* rather than the original space: the residual removes the large shared component that otherwise dominates the cosine similarity, so comparisons reflect each token's genuinely complementary information. For the i-th centered visual token $v_{c,i}$ with text-subspace reconstruction $\hat{v}_{c,i}$, the residual is

$$r_i = v_{c,i} - \hat{v}_{c,i}, \tag{11}$$

   i.e., the residual after regularized reconstruction from the text representations. Redundancy between two tokens is measured by the squared cosine similarity of their residuals,

$$g_{ij} = \left(\frac{r_i^\top r_j}{\|r_i\|_2 \, \|r_j\|_2} \right)^2 \in [0, 1], \tag{12}$$

   where a larger g_{ij} means more similar residual directions and thus higher redundancy.

2. **Diversity-Aware Greedy Selection.** Based on the guided scores and residual similarities, SOAR iteratively selects informative and non-redundant visual tokens. At each step, it selects

$$i^* = \arg\max_{i \notin \mathcal{S}} \mathrm{Score}_i, \tag{13}$$

   with $\mathcal{S}$ the set of selected tokens, and then discounts the remaining candidates by their residual similarity to i^*:

$$\mathrm{Score}_j \leftarrow \mathrm{Score}_j (1 - g_{i^* j}), \quad j \notin \mathcal{S}. \tag{14}$$

   Since $g_{i^* j} \in [0, 1]$ is a dimensionless quantity normalized by the residual magnitudes, the factor $1 - g_{i^* j}$ is a scaling coefficient in $[0, 1]$ that attenuates Score_j proportionally without altering its dimension. Candidates whose residual directions align with the selected token are thereby progressively suppressed, and the process continues until the budget K is reached.

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUP

Models and Baselines. We evaluate SOAR on three VLMs of diverse architectures, including LLaVA-1.5 (Liu et al., 2023), LLaVA-NeXT (Liu et al., 2024a), and Qwen2.5-VL (Bai et al., 2025).

Datasets. We report image-based results on eight standard benchmarks: GQA (Hudson & Manning, 2019), MMBench and MMBench-CN (Liu et al., 2024b), MME (Fu et al., 2025), POPE (Li et al., 2023b), SQA (Lu et al., 2022), VQAv2 (Goyal et al., 2017), and TextVQA (Singh et al., 2019).

Table 1: Performance comparison of different token reduction methods on LLaVA-1.5-7B.

Methods	Venue	GQA	MMB	MMB-cn	MME	POPE	SQA	VQAv2	TextVQA	Average
LLaVA-1.5-7B		**Upper Bound, 576 Tokens**								
Vanilla	–	61.9	64.7	58.1	1862	85.9	69.5	78.5	58.2	**100%**
LLaVA-1.5-7B		**Retain 192 Tokens (33.3%)**								
FastV	ECCV'24	52.7	61.2	57.0	1612	64.8	67.3	67.1	52.5	89.0%
SparseVLM	ICML'25	59.5	64.1	53.7	1787	85.3	68.7	75.6	57.8	97.2%
DART	EMNLP'25	60.0	63.6	57.0	1856	82.8	69.8	76.7	57.4	98.3%
VisionZip	CVPR'25	59.3	63.0	57.3	1782	85.3	68.9	76.8	57.3	97.8%
HoloV	NeurIPS'25	59.0	65.4	58.0	1820	85.6	69.8	76.7	57.4	98.7%
SCOPE	NeurIPS'25	60.1	63.6	56.8	1804	86.4	68.8	77.2	57.7	98.3%
SpecFlow	ICML'26	58.3	65.8	56.1	1827	85.8	69.7	76.4	57.9	98.4%
SOAR	**Ours**	60.6	64.6	58.4	1820	85.5	68.9	77.8	57.8	**99.1%**
LLaVA-1.5-7B		**Retain 128 Tokens (22.2%)**								
FastV	ECCV'24	49.6	56.1	56.4	1490	59.6	60.2	61.8	50.6	83.2%
SparseVLM	ICML'25	58.4	64.5	51.1	1746	85.0	68.6	73.8	56.7	95.6%
DART	EMNLP'25	58.7	63.2	57.5	1840	80.1	69.1	75.9	56.4	97.0%
VisionZip	CVPR'25	57.6	62.0	56.7	1761.7	83.2	68.9	75.6	56.8	96.4%
HoloV	NeurIPS'25	57.7	63.9	56.5	1802	84.0	69.8	75.5	56.8	97.2%
SCOPE	NeurIPS'25	59.7	62.5	56.9	1776	86.1	68.4	76.5	57.2	97.5%
SpecFlow	ICML'26	57.6	64.3	55.9	1794	84.9	69.9	75.3	56.8	97.2%
SOAR	**Ours**	59.8	64.0	57.0	1815	85.8	68.8	77.1	57.6	**98.4%**
LLaVA-1.5-7B		**Retain 64 Tokens (11.1%)**								
FastV	ECCV'24	46.1	48.0	52.7	1256	48.0	51.1	55.0	47.8	73.6%
SparseVLM	ICML'25	53.8	60.1	52.7	1589	77.5	69.8	68.2	53.4	90.6%
DART	EMNLP'25	55.9	60.6	53.2	1765	73.9	69.8	72.4	54.4	92.8%
VisionZip	CVPR'25	55.1	60.1	55.4	1690	77.0	69.0	72.4	55.5	93.0%
HoloV	NeurIPS'25	55.3	63.3	55.1	1715	80.3	69.5	72.8	55.4	94.4%
SCOPE	NeurIPS'25	58.3	61.7	54.4	1698	83.9	68.6	75.3	56.6	95.4%
SpecFlow	ICML'26	55.3	63.7	54.3	1713	80.5	69.7	73.7	54.9	94.4%
SOAR	**Ours**	58.0	62.2	55.2	1740	83.4	69.0	75.2	56.5	**96.0%**

Table 2: Performance comparison of different token reduction methods on LLaVA-NeXT-7B.

Method	Venue	GQA	MMBench	MMBench-cn	MME	POPE	SQA	VQAv2	TextVQA	Avg
LLaVA-NeXT-7B		**Upper Bound, 2880 Tokens**								
Vanilla	–	64.2	67.4	60.6	1851	86.5	70.1	81.8	61.4	100%
LLaVA-NeXT-7B		**Retain 320 Tokens**								
FastV	ECCV'24	55.9	61.6	51.9	1661	71.7	62.8	71.9	55.7	88.1%
SparseVLM	ICML'25	56.1	60.6	54.5	1533	82.4	66.1	71.5	58.4	90.3%
VisionZip	CVPR'25	59.3	63.1	55.6	1702	82.1	67.3	76.2	58.9	93.7%
DART	EMNLP'25	61.7	65.3	58.2	1710	84.1	68.4	79.1	58.7	96.1%
HoloV	NeurIPS'25	61.7	65.3	57.5	1738	83.9	68.9	79.5	58.7	96.2%
SpecFlow	ICML'26	62.5	66.7	56.8	1707	85.0	68.6	80.1	59.5	96.6%
SOAR	**Ours**	60.9	66.5	59.6	1819	85.7	68.3	78.7	59.6	**97.5%**

5.2 Comparison Experimental Results

Results on LLaVA. Table 1 reports the results on LLaVA-1.5-7B. Here, Vanilla denotes the original model without visual-token reduction. On LLaVA-1.5-7B, SOAR achieves the best average performance among methods with complete results under all token budgets. With 192, 128, and 64 retained visual tokens, SOAR preserves 99.1%, 98.4%, and 96.0% of the Vanilla baseline, respectively. Notably, even when retaining only 64 visual tokens, SOAR still surpasses SCOPE, the strongest baseline with complete results, by 0.6 percentage points on average, demonstrating its ability to preserve critical visual information under aggressive compression.

Results on LLaVA-NeXT-7B. Table 2 presents the corresponding comparison results. SOAR likewise achieves the best reported average performance with complete results under the 320-token setting, preserving 97.5% of the Vanilla baseline. Compared with DART and HoloV, SOAR improves the average performance by 1.4 and 1.3 percentage points, respectively.

Table 3: Performance comparison of different token reduction methods on Qwen2.5-VL-7B.

Method	Venue	MMBench	MME	POPE	SQA	MMBench-cn	GQA	TextVQA	Avg
Qwen2.5-VL-7B					**Upper Bound, 1296 Tokens**				
Vanilla	–	83.4	2295	87.1	88.7	81.4	61.0	77.2	100.0%
Qwen2.5-VL-7B					**Retain 33.3% Tokens**				
FastV	ECCV'24	79.5	2306	86.9	85.6	77.9	59.5	75.0	97.5%
HoloV	NeurIPS'25	81.2	2288.9	86.1	88.7	80.0	60.2	74.4	98.5%
SOAR	**Ours**	82.5	2319	88.0	88.2	81.3	60.8	75.8	**99.7%**
Qwen2.5-VL-7B					**Retain 22.2% Tokens**				
FastV	ECCV'24	76.5	2243	85.1	84.8	75.2	57.8	73.5	95.0%
HoloV	NeurIPS'25	80.5	2268.8	85.2	88.2	79.6	59.5	71.7	97.2%
SOAR	**Ours**	81.6	2313	87.3	87.5	80.2	60.3	75.1	**98.9%**
Qwen2.5-VL-7B					**Retain 11.1% Tokens**				
FastV	ECCV'24	71.2	2061	78.6	80.6	68.2	52.4	67.0	87.5%
HoloV	NeurIPS'25	78.2	2246.5	83.7	87.6	77.1	57.5	65.7	94.4%
SOAR	**Ours**	80.0	2273	86.0	86.6	77.6	58.9	72.5	**96.7%**

These results demonstrate that SOAR generalizes effectively from the standard LLaVA-1.5 architecture to the higher-resolution LLaVA-NeXT architecture, which contains substantially more visual tokens.

Results on Qwen2.5-VL.

Table 3 reports the results on Qwen2.5-VL-7B. We fix the input resolution to 1008×1008, corresponding to 1,296 visual tokens per image. Under token retention ratios of 33.3%, 22.2%, and 11.1%, SOAR achieves 99.7%, 98.9%, and 96.7% average performance, respectively, consistently outperforming both FastV and HoloV. Notably, under the most aggressive 11.1% token budget, SOAR surpasses FastV and HoloV by 9.2 and 2.3 percentage points, respectively. This superior performance under severe compression stems from the inherently low overlap between attention relevance and CMR guidance: as the two scoring signals highlight distinct token subsets with minimal overlap, their joint fusion prevents the omission of critical visual evidence that single-criterion baselines inevitably discard. These results confirm that SOAR generalizes effectively beyond the LLaVA architecture while maintaining robust visual compression capabilities.

Table 4: Component ablation on LLaVA-1.5-7B at 11.1% visual token retention. Avg. %Full reports average performance on SQA, TextVQA, POPE, and MME relative to the full-token baseline, while ΔAvg. is measured against full SOAR.

Group	Variant	Avg. %Full	ΔAvg.
Full	**SOAR full**	**96.59**	0.00
Score	Attention score only	96.07	-0.52
Score	CMR score only	96.00	-0.59
Score	(Attention + CMR) score	96.20	-0.39
Diversity	Top-K selection	90.13	-6.46
Diversity	Raw-space diversity	95.70	-0.89

5.3 Ablation Study and Analysis

Component Ablation. To assess the contribution of each component, we ablate SOAR on LLaVA-1.5-7B under 11.1% visual token retention from two aspects: scoring signals and diversity selection, as shown in Table 4.

For scoring, using only the attention score and only the CMR score achieves 96.07% and 96.00%, respectively, showing that the geometric residual captured by CMR is itself an effective selection signal rather than an auxiliary term of attention. Combining the two signals multiplicatively gives the full SOAR result of 96.59%, outperforming either signal alone. In contrast, additive fusion reaches only 96.20%, lagging behind multiplicative fusion.

For diversity selection, direct top-k selection drops sharply to 90.13% (-6.46), indicating that selecting tokens solely by the initial scores tends to retain redundant tokens. Raw-space diversity recovers most of the performance to 95.70% (-0.89), but still underperforms full SOAR, suggesting that enforcing diversity in the residual space is more effective for suppressing redundant visual information.

Table 5: Efficiency comparison on POPE for LLaVA-NeXT-7B and LLaVA-1.5-7B.

Model	Method	Total Time	Prefill Time	KV Cache (MB)	POPE F1
LLaVA-NeXT-7B	Full	46:11	37:21	1156	86.5
	Random	15:33	7:20	192	79.1
	SparseVLM	35:55	26:57	200	82.4
	SOAR	18:32	10:20	192	85.7
LLaVA-1.5-7B	Full	16:39	9:55	319	85.9
	Random	10:49	4:10	63	75.8
	SparseVLM	12:58	6:18	66	77.5
	SOAR	11:36	4:56	63	83.4

Effectiveness Analysis of the Fusion Strategy. Attention and CMR capture complementary aspects of token importance, namely semantic relevance and geometric residual, respectively. The former focuses on regions directly related to the query, whereas the latter highlights visual regions that cannot be well explained by the text subspace. As shown in Figure 4, although both respond to informative regions, the overlap between their selected tokens remains consistently low, with a low average overlap ratio across the four benchmarks (see Appendix E for details). This indicates that the two signals are complementary rather than redundant: relying on either one alone inevitably misses tokens preferred by the other. Consequently, combining the two scores preserves a more complete set of informative visual tokens, leading to better performance than using either attention or CMR alone.

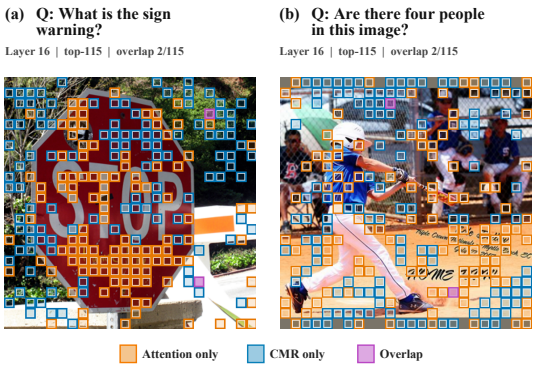


Figure 4: Selection comparison

Furthermore, multiplicative fusion is better aligned with this objective than additive fusion. Additive fusion is sensitive to differences in score magnitude, allowing one signal to dominate the final ranking. In contrast, multiplicative fusion favors tokens that simultaneously exhibit semantic relevance and substantial residual information, while naturally suppressing background or noisy tokens that receive an abnormally high score from only one signal. As a result, additive fusion achieves only 96.20%, lower than the 96.59% obtained by multiplicative fusion.

Efficiency Analysis. All token reduction methods retain 11.1% of visual tokens. Total Time and Prefill Time are wall-clock times (mm:ss) for evaluating all 8,910 POPE questions, measured on a single NVIDIA RTX 4090 GPU with FP16 precision and batch size 1, excluding model loading. For every method, all token scoring and selection costs are included in both Prefill Time and Total Time. KV Cache is the average per-sample key-value cache size in MB. SOAR achieves a strong trade-off between efficiency and performance. As shown in Table 5, compared with the full model, SOAR achieves 2.49×/1.44× total runtime speedups and 3.62×/2.01× prefill speedups on LLaVA-NeXT/LLaVA-1.5, while reducing the KV cache from 1156MB/319MB to 192MB/63MB. Even compared with Random pruning, which approximates an ideal fast baseline, SOAR remains close in runtime, with only 19%/7% additional total time, while improving POPE F1 by 6.6/7.6 points. Compared with SparseVLM, SOAR is faster on both models in total runtime and prefill time while maintaining higher accuracy.

6 Conclusion

In this work, we reveal cross-modal subspace overlap in VLMs and propose CSO for quantification. We design CMR via Tikhonov regularized subspace projection and develop the training-free SOAR to compress visual tokens. Extensive evaluations over LLaVA, Qwen and LLaVA-NeXT verify that SOAR substantially accelerates inference without obvious performance loss.

AI USE STATEMENT

In this work, we used generative AI tools to assist with language editing, including improving grammar, clarity, and presentation. We did not use generative AI tools to generate research ideas, develop the methodology, analyze results, write code, create figures, or identify citations. All AI-assisted text was carefully reviewed by the authors. We take full responsibility for the final content of this work, including any text produced with the aid of generative AI.

REFERENCES

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736, 2022.

Saeed Ranjbar Alvar, Gursimran Singh, Mohammad Akbari, and Yong Zhang. Divprune: Diversity-based visual token pruning for large multimodal models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 9392–9401, 2025.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A frontier large vision-language model with versatile abilities. *arXiv preprint arXiv:2308.12966*, 1(2):3, 2023.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your vit but faster. In *The Eleventh International Conference on Learning Representations*, 2023.

Junjie Chen, Xuyang Liu, Zichen Wen, Yiyu Wang, Siteng Huang, and Honggang Chen. Variation-aware vision token dropping for faster large vision-language models, 2026. URL https://arxiv.org/abs/2509.01552.

Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. In *European Conference on Computer Vision*, pp. 19–35. Springer, 2024a.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 24185–24198, 2024b.

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale N Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning. *Advances in neural information processing systems*, 36:49250–49267, 2023.

Jinhong Deng, Wen Li, Joey Tianyi Zhou, and Yang He. Scope: Saliency-coverage oriented token pruning for efficient multimodel llms. *arXiv preprint arXiv:2510.24214*, 2025.

Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. Palm-e: An embodied multimodal language model. *arXiv preprint arXiv:2303.03378*, 2023.

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, et al. Mme: A comprehensive evaluation benchmark for multimodal large language models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2025.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6904–6913, 2017.

Shaohan Huang, Li Dong, Wenhui Wang, Yaru Hao, Saksham Singhal, Shuming Ma, Tengchao Lv, Lei Cui, Owais Khan Mohammed, Barun Patra, et al. Language is not all you need: Aligning perception with language models. *Advances in Neural Information Processing Systems*, 36: 72096–72109, 2023.

Wenxuan Huang, Zijie Zhai, Yunhang Shen, Shaosheng Cao, Fei Zhao, Xiangfeng Xu, Zheyu Ye, and Shaohui Lin. Dynamic-llava: Efficient multimodal large language models via dynamic vision-language context sparsification. In *International Conference on Learning Representations*, volume 2025, pp. 69927–69955, 2025.

Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6700–6709, 2019.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, et al. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pp. 19730–19742. PMLR, 2023a.

KunChang Li, Yinan He, Yi Wang, Yizhuo Li, Wenhai Wang, Ping Luo, Yali Wang, Limin Wang, and Yu Qiao. Videochat: Chat-centric video understanding. *Science China Information Sciences*, 68(10):200102, 2025a.

Wentong Li, Yuqian Yuan, Jian Liu, Dongqi Tang, Song Wang, Jie Qin, Jianke Zhu, and Lei Zhang. Tokenpacker: Efficient visual projector for multimodal llm. *International Journal of Computer Vision*, 133(10):6794–6812, 2025b.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. In *The 2023 Conference on Empirical Methods in Natural Language Processing*, 2023b.

Bin Lin, Yang Ye, Bin Zhu, Jiaxi Cui, Munan Ning, Peng Jin, and Li Yuan. Video-llava: Learning united visual representation by alignment before projection. In *Proceedings of the 2024 conference on empirical methods in natural language processing*, pp. 5971–5984, 2024.

Zhihang Lin, Mingbao Lin, Luxi Lin, and Rongrong Ji. Boosting multimodal large language models with visual tokens withdrawal for rapid inference. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 5334–5342, 2025.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023.

Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llavanext: Improved reasoning, ocr, and world knowledge, 2024a.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. Mmbench: Is your multi-modal model an all-around player? In *European conference on computer vision*, pp. 216–233. Springer, 2024b.

Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. *Advances in Neural Information Processing Systems*, 35:2507–2521, 2022.

Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Khan. Video-chatgpt: Towards detailed video understanding via large vision and language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, pp. 12585–12602, 2024.

Yuzhang Shang, Mu Cai, Bingxin Xu, Yong Jae Lee, and Yan Yan. Llava-prumerge: Adaptive token reduction for efficient large multimodal models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 22857–22867, 2025.

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8317–8326, 2019.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

Zichen Wen, Yifeng Gao, Shaobo Wang, Junyuan Zhang, Qintong Zhang, Weijia Li, Conghui He, and Linfeng Zhang. Stop looking for important tokens in multimodal language models: Duplication matters more. *arXiv preprint arXiv:2502.11494*, 2025.

Long Xing, Qidong Huang, Xiaoyi Dong, Jiajie Lu, Pan Zhang, Yuhang Zang, Yuhang Cao, Conghui He, Jiaqi Wang, Feng Wu, et al. Pyramiddrop: Accelerating your large vision-language models via pyramid visual redundancy reduction. *arXiv preprint arXiv:2410.17247*, 2024.

Cheng Yang, Yang Sui, Jinqi Xiao, Lingyi Huang, Yu Gong, Chendi Li, Jinghua Yan, Yu Bai, Ponnuswamy Sadayappan, Xia Hu, et al. Topv: Compatible token pruning with inference time optimization for fast and low-memory multimodal vision language model. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 19803–19813, 2025a.

Senqiao Yang, Yukang Chen, Zhuotao Tian, Chengyao Wang, Jingyao Li, Bei Yu, and Jiaya Jia. Visionzip: Longer is better but not necessary in vision language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 19792–19802, 2025b.

Qinghao Ye, Haiyang Xu, Guohai Xu, Jiabo Ye, Ming Yan, Yiyang Zhou, Junyang Wang, Anwen Hu, Pengcheng Shi, Yaya Shi, et al. mplug-owl: Modularization empowers large language models with multimodality. *arXiv preprint arXiv:2304.14178*, 2023.

Haokui Zhang, Congyang Ou, Dawei Yan, Peng Wang, Qingsen Yan, Yu Zhang, Ying Li, and Rong Xiao. Trio: Token reduction via inference-objective guidance for efficient vision-language models, 2026. URL https://arxiv.org/abs/2602.04657.

Qizhe Zhang, Mengzhen Liu, Lichen Li, Ming Lu, Yuan Zhang, Junwen Pan, Qi She, and Shanghang Zhang. Beyond attention or similarity: Maximizing conditional diversity for token pruning in mllms. *arXiv preprint arXiv:2506.10967*, 2025a.

Yuan Zhang, Chun-Kai Fan, Junpeng Ma, Wenzhao Zheng, Tao Huang, Kuan Cheng, Denis A Gudovskiy, Tomoyuki Okuno, Yohei Nakata, Kurt Keutzer, and Shanghang Zhang. SparseVLM: Visual token sparsification for efficient vision-language model inference. In *Forty-second International Conference on Machine Learning*, 2025b.

Deyao Zhu, Xiaoqian Shen, Xiang Li, Mohamed Elhoseiny, et al. Minigpt-4: Enhancing vision-language understanding with advanced large language models. In *International Conference on Learning Representations*, volume 2024, pp. 18378–18394, 2024.

Xin Zou, Di Lu, Yizhou Wang, Yibo Yan, Yuanhuiyi Lyu, Xu Zheng, Linfeng Zhang, and Xuming Hu. Don't just chase" highlighted tokens" in mllms: Revisiting visual holistic context retention. *arXiv preprint arXiv:2510.02912*, 2025.

APPENDIX

A ADDITIONAL VISUALIZATION OF CROSS-MODAL SUBSPACE OVERLAP

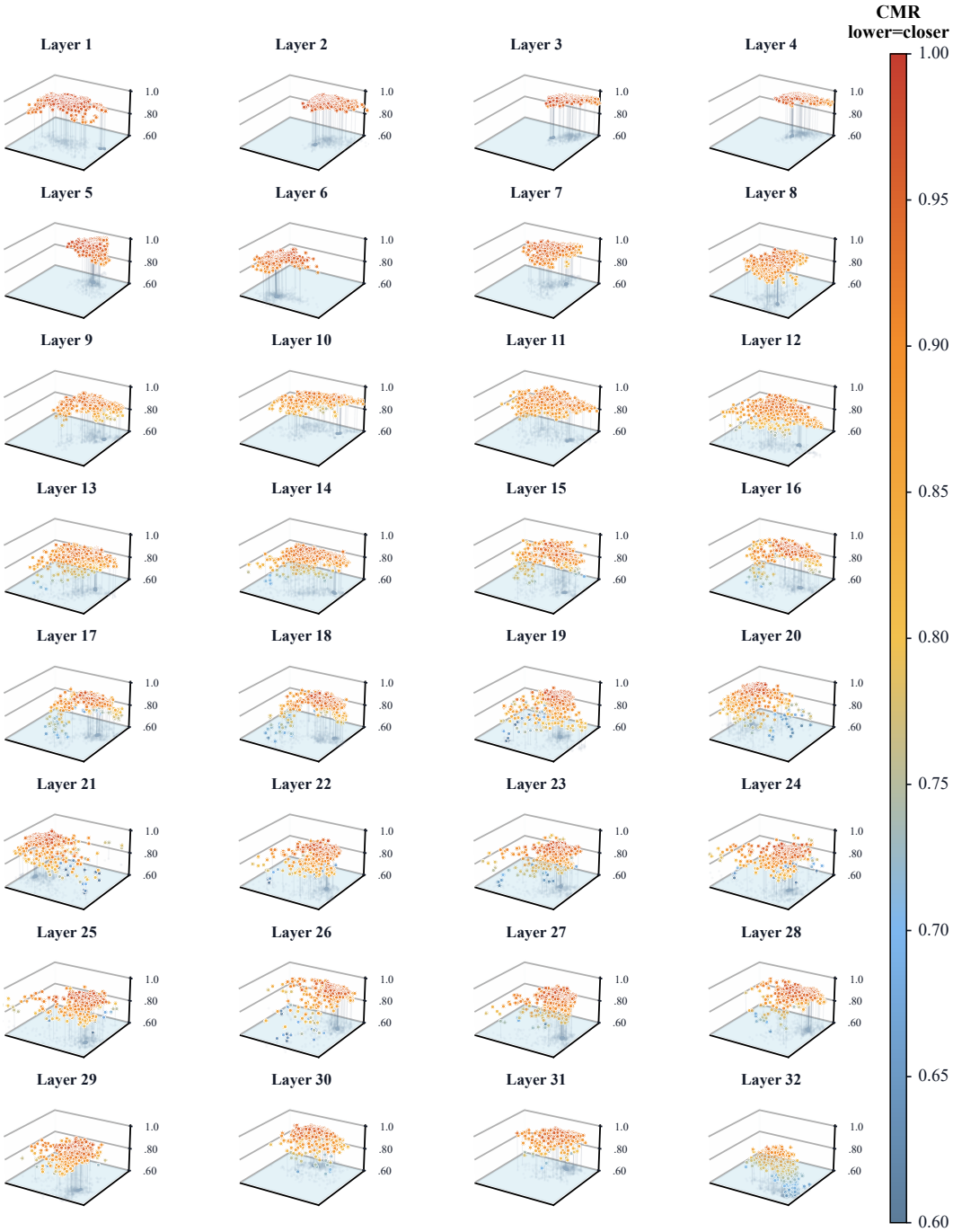


Figure 5: Full visualization of visual tokens progressively moving closer to the text subspace across all 32 layers. Each subplot corresponds to one Transformer layer. The point positions indicate the projections of visual tokens onto the text PCA plane, while the height and color represent CMR.

Figure 5 provides the complete visualization over all 32 layers, offering a more detailed view of how visual tokens evolve during layer-wise forward propagation. Overall, visual tokens progressively

converge towards the text subspace during layer-wise forward propagation. As the layer depth increases, a large number of visual tokens gradually exhibit lower CMR and form a more concentrated distribution near the text PCA plane. This suggests that the model continuously transforms visual representations into a subspace that becomes increasingly aligned with text representations in deeper layers.

Meanwhile, this process does not occur uniformly across all visual tokens, but instead exhibits a highly non-uniform and sparse structure. Most visual tokens gradually become redundant with the text subspace, meaning that their information can be well explained by text-related representations. However, a small subset of tokens still maintains relatively high CMR even in deeper layers, indicating that they are not fully explained by the text subspace and may preserve unique visual information. This phenomenon supports our key observation: visual information in multimodal models does not disappear synchronously as a whole, but its subspace overlap with text representations emerges selectively layer by layer, where most tokens gradually become redundant while a few tokens remain complementary.

B Mathematical Derivation: The Scale of CSO and the Random Baseline

Section Analysis shows that the final-layer CSO value is usually around 0.2–0.4. Although this value may appear small under the naive scale of $[0, 1]$, it should be interpreted relative to the dimensionality of the text subspace. Since the number of text tokens is much smaller than the hidden dimension, the text subspace is intrinsically low-dimensional and cannot explain all visual information, even in the most favorable case.

This appendix provides a simple derivation of the effective scale of CSO. We first discuss the maximum CSO that a fixed low-dimensional subspace can achieve, then derive the random baseline when visual and text representations are independent, and finally compare these quantities with empirical observations.

B.1 Upper Bound of CSO Under a Fixed Subspace Dimension

Let $V_{\text{vis}} \in \mathbb{R}^{N_v \times D}$ denote the visual-token matrix at a given layer, where N_v is the number of visual tokens and D is the hidden dimension. Let $U \in \mathbb{R}^{D \times k}$ be an orthonormal basis of a k-dimensional subspace, satisfying $U^\top U = I_k$. The projection of visual tokens onto this subspace is

$$\text{proj}_U(V_{\text{vis}}) = V_{\text{vis}} U U^\top. \tag{15}$$

The corresponding CSO is defined as the fraction of visual information that can be explained by this subspace:

$$\begin{aligned} \text{CSO}(U) &= 1 - \frac{\|V_{\text{vis}} - V_{\text{vis}} U U^\top\|_F^2}{\|V_{\text{vis}}\|_F^2} \\ &= \frac{\|V_{\text{vis}} U\|_F^2}{\|V_{\text{vis}}\|_F^2}. \end{aligned} \tag{16}$$

Here, the denominator denotes the total amount of visual information, while the numerator measures the part that lies in the subspace spanned by U.

To understand the maximum possible value of CSO, we decompose the visual representation into its principal directions. Let the singular values of V_{vis} be

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_R, \tag{17}$$

where $R = \min(N_v, D)$. The total visual information can then be written as

$$\|V_{\text{vis}}\|_F^2 = \sum_{r=1}^{R} \sigma_r^2. \tag{18}$$

Intuitively, σ_r^2 measures how much information is contained in the r-th principal direction. Therefore, if we can only use a k-dimensional subspace to explain the visual representation, the best possible choice is to align this subspace with the top-k principal directions of V_{vis}. These directions contain the largest amount of visual information.

Thus, for any k-dimensional subspace U, CSO is upper-bounded by

$$\text{CSO}(U) \leq \frac{\sigma_1^2 + \sigma_2^2 + \cdots + \sigma_k^2}{\sigma_1^2 + \sigma_2^2 + \cdots + \sigma_R^2}. \tag{19}$$

The equality holds only when the subspace U is exactly aligned with the top-k principal directions of V_{vis}.

This result indicates that the upper bound of CSO is not 1.0. Instead, it is determined by how much visual information is contained in the top-k principal directions. In other words, a low-dimensional subspace is not able to explain all visual information.

In LLaVA-1.5-7B, the number of post-image text tokens is typically about $N_t \approx 30$, while the hidden dimension is $D = 4096$. Therefore, the text subspace is very low-dimensional. Empirically, the top 30 principal directions of visual representations usually account for about 40%–70% of the total visual information. As a result, even if the text subspace were perfectly aligned with the dominant visual directions, the theoretical upper bound of CSO would be around 0.4–0.7, rather than 1.0.

This explains why a final-layer CSO value of 0.2–0.4 should not be considered small. It already occupies a substantial portion of the effective range that a low-dimensional text subspace can reach.

B.2 Random Baseline Under Independent Visual and Text Representations

We next consider the opposite case, where visual and text representations have no systematic alignment. In this case, the text subspace can be regarded as a random k-dimensional subspace in a D-dimensional space.

For a random k-dimensional subspace, the expected fraction of any vector that falls into this subspace is approximately

$$\frac{k}{D}. \tag{20}$$

Therefore, when visual and text representations are independent, the expected CSO is

$$\mathbb{E}[\text{CSO}] = \frac{k}{D} \approx \frac{N_t}{D}, \tag{21}$$

where $k = N_t$ corresponds to the effective dimension of the text subspace.

For LLaVA-1.5-7B, we have $N_t \approx 30$ and $D = 4096$, which gives

$$\frac{N_t}{D} \approx \frac{30}{4096} \approx 0.0073. \tag{22}$$

Empirically, the layer-0 CSO values are 0.008 on POPE, 0.009 on MME, and 0.017 on TextVQA. The values on POPE and MME are very close to the random baseline, while the value on TextVQA is slightly higher, possibly because this task already induces mild visual-text alignment in shallow layers.

Overall, the shallow-layer CSO values are of the same order as the random baseline. This suggests that visual and text representations are approximately independent in shallow layers, and that meaningful cross-modal subspace overlap has not yet emerged.

B.3 The Effective Scale for Interpreting CSO

Combining the upper-bound analysis and the random-baseline analysis, CSO should not be interpreted directly on the naive scale of $[0, 1]$. Instead, its effective scale is approximately

$$\left[\frac{N_t}{D}, \frac{\sigma_1^2 + \cdots + \sigma_{N_t}^2}{\sigma_1^2 + \cdots + \sigma_R^2}\right] \approx [0.007,\ 0.4\text{–}0.7]. \tag{23}$$

The left endpoint corresponds to the random baseline, where visual and text representations are nearly independent. The right endpoint corresponds to the ideal case, where the text subspace is perfectly aligned with the dominant visual directions.

Table 6 shows representative CSO values on POPE for LLaVA-1.5-7B.

Table 6: CSO at representative layers of LLaVA-1.5-7B on POPE, together with their relative positions on the effective scale.

Layer	CSO	vs. Baseline	vs. Bound (0.5)
0	0.008	$\approx 1\times$	1.6%
15	0.035	$4.8\times$	7%
25	0.095	$13\times$	19%
31	0.209	$29\times$	42%

As shown in the table, the CSO value at layer 0 is almost the same as the random baseline. In contrast, the final-layer CSO is about $29\times$ larger than the random baseline and reaches around 42% of the theoretical upper bound.

Therefore, although the absolute CSO value at the final layer is only 0.209, it is already far above random alignment and occupies a considerable portion of the attainable range of a low-dimensional text subspace. This supports our observation that visual tokens are progressively aligned to text subspace during the forward propagation of VLMs, revealing a systematic cross-modal phenomenon inside the model.

C FlashAttention-Compatible Attention Scoring

SOAR uses an attention signal to estimate how strongly post-image text tokens attend to visual tokens. For the h-th attention head, let its causal attention matrix be

$$P^h = \mathrm{Softmax}\left(\frac{Q^h (K^h)^\top}{\sqrt{d_h}} + M\right), \tag{24}$$

where M denotes the causal attention mask and d_h is the head dimension. Let $\mathcal{V}$ denote the set of visual-token positions and T_{post} denote the set of post-image text-token positions. The attention score required by SOAR is defined as

$$a_i = \sum_{h \in H_{\text{top}}} \sum_{t \in T_{\text{post}}} P_{t,i}^h, \quad i \in \mathcal{V}. \tag{25}$$

This score measures how much visual token i is attended to by post-image text tokens at the current layer. A direct implementation would require explicitly accessing the submatrix $P^h_{T_{\text{post}},\mathcal{V}}$. However, FlashAttention does not materialize the full $N \times N$ attention matrix, but directly computes the attention output in a block-wise streaming manner:

$$O^h = P^h V^h. \tag{26}$$

Therefore, explicitly reading $P_{t,i}^h$ is incompatible with the memory-efficient design of FlashAttention.

To avoid materializing P^h, SOAR adopts a forward-only dual-flash aggregation strategy to compute the required text-to-visual attention statistics. The idea is related to the dual-flash operation used in

SparseVLM (Zhang et al., 2025b), but the auxiliary value matrix is designed differently. SparseVLM typically marks selected text raters in the auxiliary value matrix to obtain token-to-rater attention statistics. In contrast, SOAR requires the attention from post-image text queries to visual keys; therefore, the auxiliary value matrix should mark visual-token positions.

Let the visual-token positions be

$$\mathcal{V} = \{v_1, v_2, \ldots, v_{N_v}\}. \tag{27}$$

We construct an auxiliary value matrix $\widetilde{V} \in \mathbb{R}^{N \times N_v}$ whose j-th row is defined as

$$\widetilde{V}[j,:] = \begin{cases} e_k, & j = v_k,\ v_k \in \mathcal{V}, \\ \mathbf{0}, & \text{otherwise}, \end{cases} \tag{28}$$

where e_k is the k-th standard basis vector. Thus, each visual token is assigned an independent marker channel, while all non-visual tokens are assigned zero auxiliary values. Using the same Q^h and K^h, we compute the auxiliary attention output as

$$\widetilde{O}^h = P^h \widetilde{V}. \tag{29}$$

Since $\widetilde{V}$ is nonzero only at visual-token positions, for any post-image text token $t \in T_{\text{post}}$, we have

$$\widetilde{O}^h_t = \sum_{j=1}^{N} P^h_{t,j} \widetilde{V}[j,:] = \sum_{k=1}^{N_v} P^h_{t,v_k} e_k. \tag{30}$$

Therefore, the k-th dimension of $\widetilde{O}^h_t$ is exactly the attention value P^h_{t,v_k}. Aggregating over all post-image text tokens gives the text-to-visual attention received by each visual token under head h:

$$g^h_{v_k} = \sum_{t \in T_{\text{post}}} \widetilde{O}^h_{t,k} = \sum_{t \in T_{\text{post}}} P^h_{t,v_k}. \tag{31}$$

This is equivalent to explicitly extracting $P^h_{T_{\text{post}},\mathcal{V}}$ and summing over the text dimension, but it only requires computing $P^h \widetilde{V}$ and never materializes the full attention matrix.

After obtaining the per-head visual attention scores, SOAR performs head selection. For each head, we compute its total visual attention mass as

$$m_h = \sum_{i \in \mathcal{V}} g^h_i. \tag{32}$$

A larger m_h indicates that the post-image text tokens in this head attend more strongly to the visual region, making the head more suitable for estimating visual-token relevance. SOAR selects the top image-attending heads:

$$H_{\text{top}} = \text{TopK}_h(m_h), \tag{33}$$

and aggregates the final visual-token attention score over the selected heads:

$$a_i = \sum_{h \in H_{\text{top}}} g^h_i. \tag{34}$$

The resulting attention score a_i is then combined with the CMR score for residual-space diversity selection.

Importantly, this auxiliary computation does not modify the normal Transformer forward pass. The main forward pass still computes the hidden states using the original value matrix:

$$O^h = P^h V^h. \tag{35}$$

The auxiliary forward pass is only used to obtain the attention statistics required for pruning:

$$\widetilde{O}^h = P^h \widetilde{V}. \tag{36}$$

Thus, SOAR does not explicitly materialize the full attention matrix or alter the SOAR FlashAttention computation. By rewriting text-to-visual attention aggregation as $P^h \widetilde{V}$, SOAR preserves post-image-text-guided attention scoring while remaining compatible with the streaming computation of FlashAttention.

D Ablation Study

D.1 Detailed Hyperparameter Sensitivity

We provide a more detailed hyperparameter sensitivity analysis of SOAR under the 11.1% visual-token retention setting on LLaVA-1.5-7B. The sweep covers two key hyperparameters: the energy ratio η for Tikhonov-regularized text-subspace projection, and the number of selected image-attending heads H_{top} for attention-score aggregation. For each configuration, we compute the average percentage over the full-token baseline across SQA, TextVQA, POPE, and MME.

Table 7 summarizes the effect of H_{top}. For each value of H_{top}, we report the mean, best, and worst average performance over all tested energy ratios. The results show that selecting too few heads leads to slightly lower performance, suggesting that overly aggressive head filtering may miss useful visual relevance signals. The best mean performance is achieved when $H_{\text{top}} = 12$, indicating that a moderate number of image-attending heads provides a good trade-off between capturing visual relevance and filtering noisy heads.

Table 7: Sensitivity to the number of selected image-attending heads H_{top}. For each H_{top}, we report the mean, best, and worst Avg. %Full over all tested energy ratios.

H_{top}	Mean over η	Best over η	Worst over η
1	95.98	96.13	95.82
2	96.04	96.19	95.93
4	96.04	96.21	95.79
8	96.20	96.37	96.05
12	**96.42**	**96.59**	96.29
16	96.24	96.51	96.05
20	96.24	96.44	96.12
24	96.32	96.52	96.05
28	96.33	96.51	96.19
32	96.27	96.52	96.05

D.2 Effect of Energy Ratio

Table 8 summarizes the effect of the energy ratio η. For each value of η, we report the mean, best, and worst average performance over all tested values of H_{top}. The mean performance varies only mildly across different energy ratios, showing that SOAR is robust to the exact threshold used to determine the Tikhonov regularization scale. Although the best single configuration appears at $\eta = 0.75$, the overall mean performance remains stable across a wide range of values, confirming that the CMR computation does not rely on a carefully tuned energy threshold.

Overall, these results demonstrate that SOAR is robust to both hyperparameters. The number of selected image-attending heads has a slightly larger effect, where moderate head selection performs best. In contrast, the energy ratio has only a mild influence, suggesting that the Tikhonov-regularized CMR computation is stable across a broad range of subspace energy thresholds. This supports that the effectiveness of SOAR mainly comes from cross-modal residual guidance and residual-space token selection, rather than from delicate hyperparameter tuning.

E Complementarity between Attention and CMR

To verify that attention and CMR provide complementary rather than redundant token selection cues, we measure the overlap between the token subsets selected by the two scores before fusion. For a given layer and token budget k, let $\mathcal{S}_k^{\text{attn}}$ and $\mathcal{S}_k^{\text{CMR}}$ denote the top-k visual token sets selected by the attention score and the CMR score, respectively. We define the overlap coverage as

$$\text{Cov@}k = \frac{\left|\mathcal{S}_k^{\text{attn}} \cap \mathcal{S}_k^{\text{CMR}}\right|}{k}. \tag{37}$$

A lower $\text{Cov@}k$ indicates that the two scores prefer more different token subsets.

Table 8: Sensitivity to the energy ratio η used in the Tikhonov-regularized projection. For each η, we report the mean, best, and worst Avg. %Full over all tested values of H_{top}.

Energy Ratio η	Mean over H_{top}	Best over H_{top}	Worst over H_{top}
0.05	96.15	96.42	95.94
0.10	96.15	96.42	95.93
0.20	96.12	96.31	95.90
0.40	96.18	96.42	95.79
0.60	96.23	96.49	96.05
0.75	96.18	**96.59**	96.02
0.85	96.22	96.48	96.02
0.90	96.23	96.38	96.03
0.95	**96.29**	96.47	96.06
0.98	96.28	96.52	95.82
0.99	96.25	96.51	95.88

Table 9: Average overlap of visual token selections by attention and CMR across all 32 layers. All values are percentages.

Benchmark	Cov@22	Cov@54	Cov@92	Cov@115	Avg.
MME	11.65	12.67	13.48	15.19	13.25
TextVQA	3.69	6.71	8.29	10.24	7.24
POPE	14.06	13.08	14.40	15.27	14.20
SQA	12.93	9.14	11.38	12.39	11.46
Average	10.58	10.40	11.89	13.27	11.54

Table 9 summarizes the average coverage over all 32 layers on four benchmarks. The overlap remains consistently low across different token budgets. The average coverage over all settings is only 11.54%, and remains 13.27% even under the larger top-115 budget. This suggests that attention and CMR are not redundant ranking signals. Attention mainly reflects query-conditioned semantic relevance, while CMR captures visual tokens with distinctive residual information that cannot be well explained by the text subspace.

This low overlap further explains why using either score alone is suboptimal. Attention-only selection may miss geometrically distinctive visual tokens with weak semantic saliency, while CMR-only selection may miss query-relevant tokens whose residual information is less prominent. Therefore, fusing the two scores covers a more complete set of useful visual evidence and leads to better performance than either single score. Meanwhile, multiplicative fusion is not equivalent to a hard intersection of two top-k sets. Instead, it performs joint ranking over continuous scores: it retains tokens with both non-trivial semantic relevance and residual information, while suppressing background or noisy tokens whose support from either signal is close to zero. This also explains its advantage over additive fusion.

F Complexity Analysis and Efficiency Discussion

F.1 Complexity and Net Acceleration Condition

For an input sequence of length n, hidden dimension d, and FFN intermediate dimension m, the computational cost of a standard Transformer layer can be approximated as

$$F(n) = 4nd^2 + 2n^2d + 2ndm. \tag{38}$$

Here, $4nd^2$ comes from the Q/K/V/O linear projections, $2n^2d$ comes from self-attention, and $2ndm$ comes from the FFN module. Therefore, reducing the sequence length decreases not only the linear projection and FFN costs, but also the quadratic self-attention cost.

If pruning reduces the sequence length from n to n', the per-layer computational saving is

$$F(n) - F(n') = (n - n')\big[4d^2 + 2d(n + n') + 2dm\big]. \tag{39}$$

This shows that the saving depends on both the number of removed tokens, $n - n'$, and the original sequence scale, $n + n'$. Thus, token pruning becomes more beneficial when the input sequence is longer and visual tokens occupy a large portion of the sequence.

SOAR introduces additional computation only at a few pruning layers. At a pruning layer, let N_v be the number of visual tokens, N_t be the number of post-image text tokens, and k be the number of retained visual tokens. The scoring cost of SOAR, including CMR projection and residual-space diversity selection, can be summarized as

$$C_{\text{score}} = O\big(N_t^2 d + N_t^3 + N_v N_t d + N_v N_t^2 + N_v^2 d + k N_v\big). \tag{40}$$

The first two terms come from text-subspace construction and Tikhonov-regularized solving, the next two terms come from projecting visual tokens onto the text subspace, and the last two terms come from residual-space similarity computation and greedy diversity selection. Since N_t is usually much smaller than N_v and d, the dominant cost is typically the residual-space Gram matrix:

$$C_{\text{score}} \approx O(N_v^2 d). \tag{41}$$

Although SOAR introduces this scoring overhead, it follows a "one-time scoring, multi-layer saving" pattern. The scoring is performed only at pruning layers, while the shortened sequence benefits all subsequent Transformer layers. Let n_0 be the original sequence length, n_l be the actual sequence length at layer l, and $\mathcal{P}$ be the set of pruning layers. SOAR achieves net acceleration when

$$\sum_{l} [F(n_0) - F(n_l)] > \sum_{p \in \mathcal{P}} C_{\text{score}}^{p}. \tag{42}$$

That is, SOAR improves inference efficiency when the accumulated saving from shorter sequences in later layers exceeds the scoring overhead introduced at pruning layers. This condition also explains why SOAR is more effective when pruning is performed earlier, more tokens are removed, and the original sequence is longer.

F.2 Discussion on Long-Sequence Inputs

For long-sequence inputs, the efficiency behavior depends on where the sequence length comes from. The first and most relevant case is long visual sequences, such as high-resolution images, multi-image inputs, or videos. In these scenarios, the number of visual tokens N_v increases, which also increases the scoring cost of SOAR, especially the $O(N_v^2 d)$ residual-space similarity computation. However, standard Transformer self-attention also contains the $O(n^2 d)$ term, and this cost is paid at every subsequent layer. Since SOAR scoring is only performed at a few pruning layers, while the benefit of shorter sequences accumulates across many later layers, the total computational saving can outweigh the additional scoring cost when pruning is applied early.

The second case is long text prompts. If the post-image text length N_t becomes very large, the cost of text-subspace construction and Tikhonov projection may increase. In this case, one can use a fixed-length post-image text window or restrict the effective rank of the text subspace, so that the terms related to N_t remain controlled.

Overall, SOAR is especially suitable for long visual-sequence scenarios where visual tokens dominate the computational cost. In such cases, the original Transformer cost grows rapidly with sequence length, while SOAR only introduces limited scoring overhead at a few pruning layers. The reduced token sequence is then reused by subsequent layers, leading to lower overall inference complexity and improved efficiency.